KENYON MICROSYSTEMS

3350 Walnut Bend • Houston, Texas 77042
Phone (713) 978-6933

SOFTWARE FOR THE HARDCORE

6800 & 6809 FORTH

t FORTH
FORTH System

t FORTH+
plus Assembler, CRT Editor

firmFORTH
produce compacted
rommable code

KENYON MICROSYSTEMS

3350 Walnut Bend • Houston, Texas 77042
Phone (713) 978-6933

FORTH is a structured, extensible programming technique which has evolved through the efforts of applications programmers, primarily the inventor Charles H. Moore of FORTH, Inc. It is designed for interactive program development and testing, and it is particularly well suited for applications which require user modification of code and interaction with data and the logic of program flow. It has both interpreter and compiler modes. Although it may be used as a programming tool within a host operating system, FORTH itself is a complete disk operating system.

The third generation 6809 microprocessor is well suited for applications ranging from controllers to high level languages for business, engineering, or scientific data analysis. It is ideal for the language FORTH. FORTH uses two stacks in memory and the 6809 has two hardware 16-bit stack pointers. Additionally, the 10 addressing modes of the 6809 include indirect and auto-increment and -decrement modes which permit easy, compact, and fast implementation of the inner workings of the FORTH kernal. The primary, most executed loop in FORTH (NEXT) requires 14 machine cycles and 4 bytes with the 6809, to be compared with 38 cycles and 14 bytes for a 6800 implementation.

The tFORTH system

The tFORTH system is an implementation of the FORTH model established by the FORTH International Standards team of the FORTH Interest Group (fig), P.O. Box 1105, San Carlos, Calif., 94070. The standard vocabulary is supplemented with additional words which permit reading and writing of FORTH text source code on disk in either standard FORTH SCREEN format or the text file format of the host disk operating system such as FLEX*; also, one may pass commands to the host system from tFORTH. It is an expanded version of 68'FORTH by R.J. Talbot. It contains the fig-FORTH text editor and a variety of words to extend data types, serial or parallel port IO, and printer control. It is available for 6800 and 6809 systems.

THE tFORTH+ AUGMENTED SYSTEM

tFORTH+ consists of tFORTH plus a set of FORTH source language code which extends the capabilities far beyond the basic fig-FORTH model. The additional extensions are:

- A full assembler using Motorola-like mnemonics instead of abstract ones found in many FORTH systems; it implements all instruction/addressing mode combinations and contains many macros for structured programming and embedding assembly code into FORTH code.
- An editor for CRT's (with greater than 68 characters per line and 18 lines) using cursor and control code character/line insert/delete/change; much faster and easier than simple fig editor.
- Source code for extended data types such as vectors, arrays, etc.
- General serial and parallel port drivers.
- Two types of CASE statements.

tFORTH provides a complete basic and useful system for learning FORTH and experimenting with applications. tFORTH+ has been designed for applications programmers who require a programming system which gives them more flexibility and power, particularly the assembler capability and easier manipulation of scientific/engineering data and instrument control.

tFORTH and tFORTH+ are available for 6800 and 6809 systems using TSC's FLEX 2.0 or FLEX 9.0. In each case the disk contains the tFORTH program in object form as a FLEX CMD file. There are start-up and demonstration files in FLEX text format. This disk also contains FORTH source language code in standard FORTH screen format. Additional disks may be all FLEX, all FORTH, or a mixture. The code is not relocatable. In addition to 8k for FLEX, the tFORTH program requires at least 16k (0000-3FFF).

Full information is supplied to permit changing the disk interface to function with another operating system, e.g., that of Smoke Signal Broadcasting.

Terminal and printer IO is easily configured to use any of the following: serial or parallel port drivers internal to FORTH, any operating system drivers (e.g., ones using video boards) or any rom monitor drivers.

THE firmFORTH SYSTEM: FIRMWARE DEVELOPMENT

firmFORTH is a package for use with tFORTH+. It consists of FORTH source code which defines a set of vocabulary words to analyse code, automatically eliminate unused code, and compact executable code by eliminating unneeded dictionary space. The purpose is to enable the efficient creation of rommable code.

The firmFORTH analysis routines trace high level words to ascertain the frequency with which lower

level words are used and flag words which require definition. These enable the user to attempt to recode to eliminate little used portions and thereby reduce memory requirements. By changing a flag, the compilation is redone, automatically eliminating words never used and dictionary portions not needed for execution. The compilation may be directed toward any target area of memory, including that containing the system itself. In the 5¼" format, a dual disk drive system is recommended, although working with one drive is possible.

firmFORTH includes in source form a basic FORTH kernal sufficient to create executable, fully self-contained, rommable code. It does not contain source code to create a new FORTH compiler or interpreter.

LICENSING AND SUB-LICENSING

tFORTH, tFORTH+, and firmFORTH are copyrighted products. The fee you pay is a one-time licensing fee that permits the use of the software on a particular computer system. Each product is identified by a serial number registered to you the user. You may make backup copies only for that computer system.

OEM's or Professional Software Systems companies who wish to create applications programs which use tFORTH or tFORTH+ may write to us for information on sub-licensing agreements. In summary, these consist of procedures to grant you the right to distribute tFORTH or tFORTH+ together with your application in source form or to distribute a product in fully compiled, non-extensible form (i.e., without compiler).

OEM's or Professional Software System companies who wish to create application programs which use only the abbreviated firmFORTH kernal, or extensions created by you, may distribute such programs to your customers with no sub-licensing fee. The only qualification is that it is to be understood by the user that Kenyon Microsystems, Inc. will continue to support only you, the sub-licensing company, and that you (our customer) must, in turn, support your customers.

*FLEX is a trademark of Technical Systems Consultants

FLEX disks and manuals

tFORTH

$100

	5¼"	8"
6800	_____	_____
6809	_____	_____

tFORTH+ (includes tFORTH plus additional FORTH source code

$ 250

	5¼"	8"
6800	_____	_____
6809	_____	_____

firmFORTH (does not include, but requires, tFORTH+ system)

$ 350

	5¼"	8"
6800	_____	_____
6809	_____	_____

Manuals alone

Vol. 1: tFORTH alone _____ $15

Vol. 2 Manual for the additional code in tFORTH+ _____ $10

Vol. 3: firmFORTH _____ $10

Payment: We accept check, VISA, Mastercharge, UPS COD, or Company Purchase Order.

ORDER NOW

Mail to:
Kenyon Microsystems, Inc.
3350 Walnut Bend
Houston, Texas 77042

Card _____ Number _____

Expiration date _____

Signature _____

Ship to: Name _____

Address _____

City _____ State _____

Country _____ Zip _____

Phone No. _____

Kenyon Microsystems, Inc.
3350 Walnut Bend
Houston, Texas 77042

!FORTH VOCABULARY

DATA TYPES: signed 16 and 32 bit integers, ARRAYS of integers, 16 bit addresses, 8 bit bytes (characters), character strings to 255 characters.

OPERATORS: + - * / */ MOD /*MOD ABS MINUS 1+ 2+ 1- 2- +- S->D 0< 0= AND OR XOR NOT < > = MIN MAX M* M/ M/MOD U/ U* D+ DMINUS DABS D+-

STACK AND MEMORY TRANSFER AND CONTROL: OVER DROP SWAP DUP -DUP ROT FILL ERASE BLANKS CMOVE @ ! C@ C! +! " "@ "!

GENERAL IO: CISER C@SER ?SER I-SER CIPAR C@PAR ?PAR I-PAR

TERMINAL AND PRINTER IO: KEY EMIT ?TERMINAL QUERY EXPECT TIB OUT COLUMNS COUNT SPACE TYPE -TRAILING PAD .LINE CR DIGIT HOLD . ? D. .R D.R #S # SIGN #> <# DPL FLD ." HEX DECIMAL BASE PORT !PRINTER !TERM +ECHO -ECHO PRINT ". "ASK LIST INDEX

FORTH DISK IO: BUFFER BLOCK BLK IN UPDATE FLUSH EMPTY-BUFFERS LOAD R/W DISKRW --> +SCR FIRST LIMIT SCR OFFSET USE PREV +BUF ;S #DR TRK/DISK SEC/TRK B/BUF SCR>BLK BLK>SCR USEBLK DRIVE DR0 DR1 DRSIM ?DISK

HOST DOS DISK IO AND COMMAND INTERFACE: DOS __ READ WRITE OPEN CLOSE REWIND FCBIN FCBOUT CLOSEIN CLOSEOUT DELETE

COMPILER WORDS AND DICTIONARY CONTROL: CONSTANT VARIABLE ARRAY CREATE <BUILDS DOES> : ; CODE ;CODE WIDTH DP HERE SMUDGE STATE CURRENT CONTEXT VOCABULARY DEFINITIONS ' C, ALLOT TRAVERSE LATEST PFA CFA NFA LFA COMPILE [COMPILE]] [?ERROR ?COMP ?EXEC ?PAIRS ?CSP ?LOADING VLIST DLIST ' MESSAGE ERROR ID. LIT CLIT TOGGLE -FIND INTERPRET NUMBER WORD

CONTROL: DO LOOP +LOOP BEGIN END UNTIL AGAIN REPEAT WHILE IF ELSE THEN ENDIF I J K LEAVE BACK BRANCH OBRANCH

MISCELLANEOUS: MON JSR EXECUTE TRACE COLD WARM +COLD +WARM SP@ SP! RP! R R> >R 0 1 2 3 BL USER +ORIGIN S0 R0 (QUIT ABORT GO NOOP FORTH LOADER DATE USRID HELP ?STACK R# DUMP TRIAD

fig-EDITOR: TEXT .S LINE WHERE #LOCATE #LEAD #LAG -MOVE TOP H E S D M T L R P I N B F X Z TILL ^I ^R CLEAR CLEAR-ALL COPY DELETE SMATCH MATCH 1LINE FIND

DISK UTILITIES: NBUFF DBUFF BMOVE SMOVE DR0>1 BACKUP0>1 V TS -TS SET-LENGTH PARTITION MAKE-NEW-DISK1